AN INTRODUCTION TO STRETCHR: THE INTELLIGENT DATASTACK

Stretchr, Inc. recently developed "Stretchr", an intelligent datastack that radically reshapes Big Data application development in a NoSQL environment.

CUSTOMER TESTIMONIAL: FRONT PORCH DIGITAL'S RAPID APPLICATION DEVELOPMENT

Front Porch Digital, Inc. is a world leader in digital asset workflow management serving global leaders in the entertainment industry. Front Porch Digital recently integrated Stretchr into its application development processes.

"Stretchr has fundamentally changed the way we approach data systems development. Today's data comes in so many shapes and sizes and is always changing, requiring you to spend a huge amount of time designing and editing schemas in traditional databases or developing expertise in NoSQL technology. With Stretchr all of that time and complexity goes away. You simply acquire the data, from any source and in any form. Stretchr then organizes the data for you based on how your users consume it – it couldn't be simpler.

Our first integration with Stretchr took an afternoon, and was effectively the insertion of one line of code into our existing application. So happy are we with the way Stretchr works and performs that we are tightly integrating our newest products with Stretchr, cutting development times significantly".

-- Phil Jackson, COO, Front Porch Digital

Contents

The NoSQL Impact on the General Purpose Database Market

BIG USERS, BIG DATA AND CLOUD COMPUTING ARE DATABASE INDUSTRY GAME-CHANGERS

The general purpose database industry has been dominated by relational database technologies for 40 years. However, Big Users, Big Data, and Cloud Computing application developers are increasingly turning to NoSQL schema-less technologies because of:

- Skyrocketing concurrent users as applications become increasingly accessible via the web (and a multitude of devices – "the internet of things").
- Soaring amounts of data collected and processed as it becomes easier and increasingly valuable to capture all kinds of data.
- An explosion of unstructured or semi-structured data - its use becomes integral to the value and richness of applications.

NoSQL provides improved scalability and performance, a relaxed consistency and an increased agility for schema-less that traditional relational technologies cannot offer.

Today, the use of NoSQL technology is rising rapidly among internet companies and enterprises as developers increasingly turn to schema-less databases for application development. Big Data application growth will explode to ~$7 billion by 2017* as firms leverage data continuously collected from wider and more diverse endpoints. Firms use this data to enrich the user experience, provide valuable tools for customers, and repurpose and monetize their information.

Organizations recognize that operating at this scale is more effectively achieved running on clusters of standard, commodity servers, and a schema-less data model for:

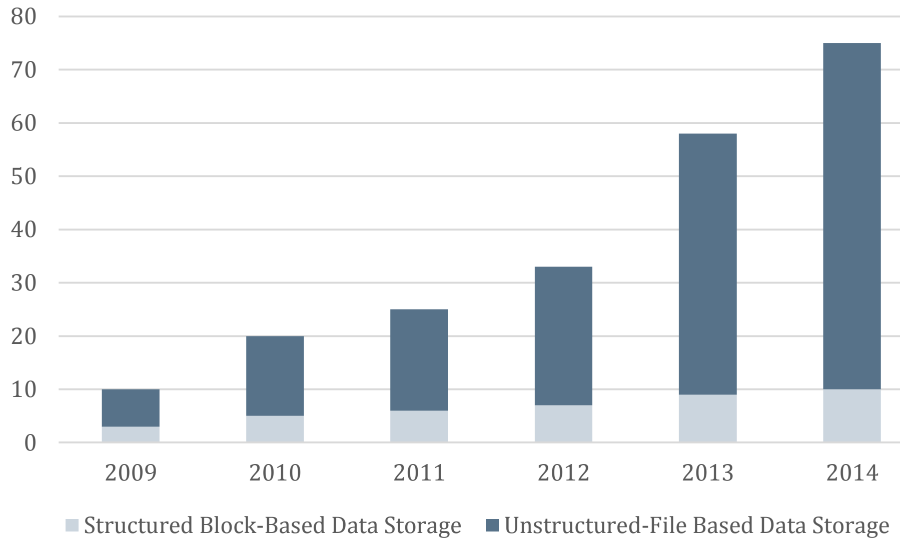
- Better application development productivity through a more flexible data model.
- Improved performance to satisfy expectations of users wanting highly responsive applications and to allow more complex data processing.
- Greater ability to scale dynamically to support more users and data.

*source: Wikibon, 451 Research

THE CHALLENGE OF NOSQL APPLICATION DEVELOPMENT

Schema-less database technology is increasingly considered a viable alternative to relational databases, especially as a more scalable approach for handling the variety of unstructured data most often captured and processed for Big Data applications today. Underscoring this trend is the rapid adoption of NoSQL database technology and the massive growth in storage from less than 10 to nearly 80 Exabytes by 2014 (see Exhibit I).

Exhibit I: 2009 - 2014 Growth in Exabytes: Structured vs. Unstructured Data Storage



Source: CSC Corporation

Leveraging this data, however, and transforming it into actionable information requires more than a NoSQL database. NoSQL technology, while providing significant scale, speed, and cost benefits, brings with it operational and skillset pain-points for developers. Specifically, what is needed is considerable NoSQL DB expertise to integrate the application, data API expertise to expose only the right pieces of data required by the application, and database administration expertise to maintain and optimize the datastore.

Stretchr was built as an intelligent datastack, by developers, for developers to address these pain points, and free up 50%-75% of a developers time, instead, to work on the application.

*source: Wikibon, 451 Research

Stretchr Overview

THE INTELLIGENT DATASTACK

Stretchr is a highly available, flexible NoSQL datastack with a powerful RESTful front-end that dramatically improves the development experience for engineers. Utilizing Stretchr alleviates the need for engineers to learn new NoSQL technologies, to rely on the skills of a data API Developer and to require the skills of a Database Administrator to build an app. In summary, Stretchr fundamentally and radically changes application development processes by eliminating the reliance of developers on other resources in the IT department.

Stretchr:

- Minimizes the time required to implement a central NoSQL datastack by eliminating the NoSQL learning curve for existing developers, and radically simplifying the onboarding process for new ones.
- Allows for exposure of a centralized datastack that powers many applications with overlapping, yet different data needs.
- Enables developers to segment, filter and view their data as needed for any given application.
- Provides relational data capabilities, via a data agnostic nesting structure, that is lost in traditional NoSQL environments.
- Addresses and alleviates data accessibility and security concerns by providing fine-grained control over what data the applications and users have access to at any given time.

Stretchr's features include a flexible architecture, a RESTful API, advanced querying and analytics, and authorization/authentication capabilities.

FLEXIBLE, HORIZONTALLY SCALABLE ARCHITECTURE

Stretchr consists of load balanced scalable API and data servers with connected disk storage. Stretchr's architecture was designed from the ground up to scale horizontally. Implementations can range from single machine development instances to multi-site, sharded and replicated server farms in physical or virtual environments. Increasing or decreasing scale is as easy as

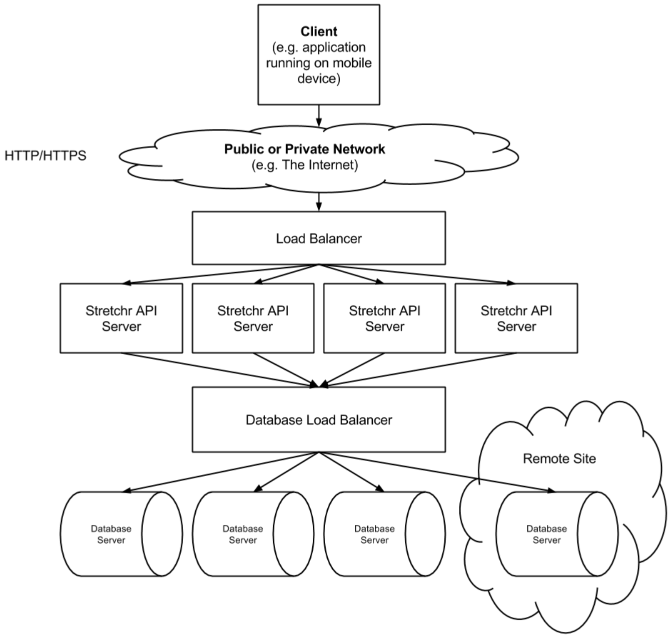
*source: Wikibon, 451 Research

provisioning or removing virtual machines. This brings flexibility to size and fine-tune the right solution for any deployment, scaling up or down on-demand without any impact to the production

*source: Wikibon, 451 Research

system (see Exhibit II below).

Exhibit II: Stretchr Architecture



NESTED RESTful INTERFACE

Stretchr's nested RESTful interface (patent pending) provides a flexible but predictable API for developers to access NoSQL data. Stretchr's nested data pattern provides powerful parent-child relationships, solving a key relational pain-point for developers migrating to NoSQL technologies while simultaneously wishing to retain relationships in their data.

The ability to query nested NoSQL data structures is one of the defining benefits of utilizing Stretchr over a traditional datastore, as it provides a view into data from different perspectives to find the best level of detail for any application. A great user example is dashboards. Customer-facing dashboards can be limited to just the data nested under the customer, while Company-facing support dashboards are given access to data nested across all customers.

Another advantage is Stretchr's suite of open-source SDKs. SDKs in Javascript, C#, Cocoa, Ruby, and Go that allow developers to program in their own language. This minimizes the effort developers require to integrate with the datastack, while providing data access from any network enabled device, such as mobile devices and browsers. Our Cocoa and JavaScript SDKs enable

*source: Wikibon, 451 Research

complex data-driven applications to be built directly against the NoSQL database, all while honoring user-defined permissions and authentication. In addition, since RESTful APIs use URLs to address resources, developers get all the benefits of the web, such as:

- Caching the same GET requests reduces the load on the system
- Dynamic routing of requests, allowing a minor URL change to control which version of Stretchr is being accessed
- Data that is addressable anywhere through a simple, static URL.

ADVANCED QUERYING

Stretchr's advanced data query capabilities enable quick filtering, ordering and paging of data directly in the URL. For example, queries can be based on:

- Whether or not a field exists.
- Whether or not an array contains an entry/object.
- Numerical comparisons: e.g. “Less than”, “greater than”, between two numbers or dates.
- Checks on deep nested objects, and objects inside arrays.
- Regular expression matching.

ANALYTICS

As Stretchr becomes the central datastack for a system, its built-in analytics and log capabilities provide invaluable insights into the usage of data throughout the entire system, for example:

- Who is using this data and why?
- What services are most active?
- What are the most common queries?
- How quickly are our queries completing and when will we need to scale up?
- What data is being used?

*source: Wikibon, 451 Research

- What data could be deprecated?
- What's our read to write ratio?
- When is the system least active, and therefore can be brought down for maintenance?
- Are we utilizing all of our allocated resources, or should we tune it down a little?

Analytics are accessible in the same nested RESTful API, allowing developers to utilize the same SDKs for interacting with Stretchr as they use to pull data from it.

AUTHENTICATION & AUTHORIZATION

Each request into Stretchr carries with it an API key which is generated by the system and given to authorized clients. An API key gives each application governable access to data resources, and is used as part of the analytics to track the activity of each client.

Stretchr's rules engine controls how much and what kind of access each client has to the data. Rules are a simple set of JSON instructions attached to an API key that define what the access is for the subsets of data requests containing the key. Some examples include:

- Customer API Keys that can be restricted to only their individual data, blocking them from accessing other customer information.
- Internal systems that have the ability to post to their own log collections, but not touch the logs of other systems.
- Data that is exposed and monetized to first, second or third parties while maintaining fine-grained control over data rights.

Stretchr's support for users is built right into the core, adding an extra layer of authorization/permissions in addition to the API Key. This enables different levels of access not only to individual applications, but individual users of the applications as well. Even though an API Key is used to identify and authorize applications/clients, users can also include another key to authorize themselves.

A user key can be mandatory or optional according to the rules created for each individual application key. Rules can be based on any content stored in the datastack. Some examples include:

*source: Wikibon, 451 Research

- Users of Key "A" can create books, but users of Key “B” can only read them.
- All users can add comments, but a user may only modify his or her own comment.
- Only users over the age of 18 may add comments or view specific content.

Use Case

ENTERPRISE DASHBOARD APPLICATION DEVELOPMENT

Dashboard applications are a notoriously challenging problem for companies with highly distributed architectures. Not only does data need to be centralized in a usable conduit, it must be secured, segmented and distributed to various parts of the business value chain. For example, customers need usage dashboards; service/support organizations require dashboards across the customers but are blocked from accessing confidential billing information; and finance departments must have access to billing information but don't require any specific customer data. These multiple utilities of customer data quickly escalates to a development nightmare.

Using Stretchr's fine-grained permissions and user authentication, however, makes segmenting, distributing, and dashboarding data an easily achievable task. Customer dashboards have a key that requires user authentication for access. Once a user is authenticated, access can be limited to only their individual subset of the data (using oauth or a custom auth solution). Keys and access rules can be created for each department in an organization, providing access only to the data specifically required for their needs.

Because authentication and authorization is being handled in the datastack, and not in a separate backend, these dashboards can be developed in any language for any mobile or desktop platform without recurring backend development.

Conclusion

The general purpose database market is undergoing a profound shift toward NoSQL technologies to accommodate the growth in Big Data scale and diversity. Through 2014 this shift will drive 80 Exabytes of storage – up from 10 EB in 2009. Big Data application growth, driven by firms analyzing, repurposing and monetizing their multi-format, multi-sourced data is estimated to be a leading growth segment of this market, from $480M in 2012 to $6.9B in 2017.

Stretchr, an intelligent datastack provides the foundation for a radically different application design capability in the NoSQL environment. This technology, built by developers for developers, enables rapid development without the time, skills and cost of a traditional, multifunctional development process. Its RESTful API and SDKs in Javascript, Ruby, Go (Java, Cocoa and C# are in development) enable developers to start programming their applications immediately without NoSQL knowledge. In addition, Stretchr's intelligent datastack feature set provides data agnostic nesting, advanced querying, analytics, and authentication/authorization that is built on a flexible infrastructure designed for horizontal scaling.

Stretchr's intelligent datastack is the platform for rapid and radical application development in the NoSQL general purpose database market.

Company Information

Stretchr, Inc.
4209 Niblick Dr.
Longmont, CO 80503
www.stretchr.com
@stretchrcom

*source: Wikibon, 451 Research